**ParteYepez**  
Introducción del proyecto

Un sistema de ventas eficiente y bien organizado es fundamental para el éxito de cualquier negocio, y esto es especialmente cierto en el caso de una empresa minorista de repuestos y accesorios para automóviles, similar a AutoZone. Estas empresas, conocidas como tiendas de autopartes, se dedican a brindar a los clientes una amplia variedad de productos automotrices, desde repuestos y aceites hasta herramientas y productos de mantenimiento.

El objetivo principal es proporcionar a los clientes una experiencia de compra rápida y conveniente, al tiempo que garantiza un servicio al cliente excepcional y una gestión eficiente del inventario. Estos sistemas están diseñados para optimizar los procesos de venta, desde el momento en que el cliente ingresa a la tienda hasta que realiza su compra y recibe sus productos.

Un componente clave es la implementación de una infraestructura tecnológica sólida. Esto implica el uso de software de punto de venta (POS) que permite a los empleados procesar transacciones de manera eficiente, mantener un registro de inventario en tiempo real y generar informes de ventas y análisis de datos. Además, estos sistemas pueden integrarse con otros sistemas empresariales, como el seguimiento de pedidos y la gestión de la cadena de suministro, para facilitar una operación fluida.

En cuanto a la gestión del inventario utiliza técnicas como la codificación de productos, la clasificación y el etiquetado adecuados para garantizar una ubicación y búsqueda eficientes de los productos en la tienda. Además, estos sistemas suelen incorporar herramientas de pronóstico de demanda y reabastecimiento automático, lo que ayuda a minimizar las existencias no vendidas y garantiza que los productos estén disponibles cuando los clientes los necesiten.

Otro aspecto crucial es la capacitación y el desarrollo del personal. Los empleados deben estar familiarizados con los productos que se ofrecen, así como con las últimas tendencias y tecnologías automotrices. Esto les permite brindar asesoramiento experto a los clientes y asegurarse de que encuentren las piezas y los accesorios adecuados para sus necesidades.

En resumen, ZoneAuto Care es una combinación cuidadosamente diseñada de tecnología, gestión de inventario y servicio al cliente para ofrecer una experiencia de compra sin problemas en una tienda de autopartes. Al implementar un sistema de este tipo, las empresas pueden mejorar la eficiencia operativa, aumentar la satisfacción del cliente y mantenerse competitivas en un mercado en constante evolución.

# Manejador de Base de datos

PhpMyAdmin es una popular herramienta de administración de bases de datos que se utiliza comúnmente en combinación con el sistema de gestión de bases de datos MySQL. La versión específica que utilizamos es la 5.7.31, que es una versión estable y ampliamente utilizada en entornos de desarrollo y producción.

PhpMyAdmin proporciona una interfaz gráfica basada en web que permite a los usuarios administrar y manipular bases de datos MySQL de manera fácil y conveniente. A través de esta herramienta, los usuarios pueden realizar diversas tareas, como crear y eliminar bases de datos, gestionar tablas, ejecutar consultas SQL, importar y exportar datos, y administrar usuarios y permisos.

Al utilizar PhpMyAdmin con MySQL versión 5.7.31, los usuarios se benefician de las características y mejoras específicas de esa versión de MySQL. Algunas de las características clave de MySQL 5.7.31 incluyen:

* Mejoras en la seguridad: MySQL 5.7.31 ofrece mejoras significativas en la seguridad, como el cifrado de contraseñas mediante el algoritmo de hash SHA- 256, la capacidad de requerir contraseñas más fuertes y el soporte para autenticación de dos factores.
* Rendimiento optimizado: Esta versión de MySQL ofrece mejoras en el rendimiento, incluida una mayor velocidad de ejecución de consultas y mejoras en el rendimiento de la replicación.
* Gestión de transacciones mejorada: MySQL 5.7.31 proporciona características mejoradas para la gestión de transacciones, incluida la compatibilidad con transacciones parciales y la capacidad de realizar rollback en transacciones parciales.
* Funciones de programación extendidas: Esta versión de MySQL ofrece nuevas funciones y mejoras en las capacidades de programación, como funciones de agregado JSON, funciones para trabajar con fechas y horas, y soporte para cursores del lado del servidor.

PhpMyAdmin simplifica la administración de una base de datos MySQL al proporcionar una interfaz intuitiva y fácil de usar para realizar tareas de administración y consulta. Los usuarios pueden navegar por las bases de datos, ver la estructura de las tablas, editar registros, ejecutar consultas SQL complejas y realizar diversas acciones de gestión de usuarios y permisos, todo ello a través de una interfaz gráfica amigable.

En resumen, PhpMyAdmin con MySQL versión 5.7.31 ofrece una forma poderosa y conveniente de administrar bases de datos MySQL. Al aprovechar las características y mejoras de MySQL 5.7.31, los usuarios pueden gestionar sus bases de datos de manera eficiente y segura a través de la interfaz intuitiva y funcional de PhpMyAdmin.

# Lenguaje de Desarrollo de la interfaz de usuario

PHP (Hypertext Preprocessor) es un lenguaje de programación ampliamente utilizado para el desarrollo web. La versión específica que mencionas es la 7.3.21, que es una versión estable y popular en la comunidad de desarrollo de PHP.

PHP 7.3.21 presenta varias mejoras y correcciones de errores en comparación con versiones anteriores. Algunas características y mejoras clave de PHP 7.3 incluyen:

* Rendimiento mejorado: PHP 7.3 introduce optimizaciones en el motor interno del lenguaje, lo que resulta en una mayor velocidad de ejecución de los scripts PHP. Esto significa que las aplicaciones PHP pueden funcionar de manera más eficiente y responder más rápidamente a las solicitudes de los usuarios.
* Soporte para heredoc y nowdoc en cadenas: Esta versión de PHP permite el uso de la sintaxis heredoc y nowdoc en cadenas, lo que simplifica la manipulación y construcción de texto de manera más legible y estructurada.
* Soporte para JSON\_THROW\_ON\_ERROR: PHP 7.3 introduce la opción JSON\_THROW\_ON\_ERROR al decodificar o codificar JSON. Esto permite que las funciones JSON de PHP arrojen excepciones en lugar de errores silenciosos, lo que facilita la detección y manejo de errores al trabajar con datos JSON.
* Extensiones actualizadas: PHP 7.3 incluye actualizaciones en varias extensiones, como OpenSSL, cURL y SQLite, lo que proporciona mejoras de seguridad y funcionalidad en esas áreas.

Es importante tener en cuenta que PHP 7.3.21 es una versión en el ramal 7.3.x de PHP y se considera una versión de soporte a largo plazo (LTS, por sus siglas en inglés). Esto significa que se proporcionan correcciones de errores y actualizaciones de seguridad para esta versión durante un período extendido, lo que la convierte en una opción confiable para aplicaciones y proyectos que requieren estabilidad y compatibilidad a largo plazo.

En resumen, PHP 7.3.21 es una versión estable y popular de PHP que ofrece mejoras de rendimiento, nuevas características y correcciones de errores en comparación con versiones anteriores. Al utilizar esta versión, los desarrolladores pueden aprovechar las mejoras de rendimiento y las características actualizadas para crear aplicaciones web eficientes y seguras.

# Listado de requisitos para la instalación del manejador de base de datos

Primeramente, para el apartado del manejador de base de datos necesitaras lo siguiente: Hardware

1. Sistema operativo: phpMyAdmin es compatible con una amplia gama de sistemas operativos, incluidos Windows, macOS y varias distribuciones de Linux. Asegúrate de que tu sistema operativo cumpla con los requisitos mínimos para phpMyAdmin.
2. Espacio en disco: phpMyAdmin en sí mismo no requiere mucho espacio en disco. Sin embargo, necesitarás espacio suficiente para alojar tu servidor web, la base de datos MySQL y cualquier otro componente necesario para tu aplicación web.
3. Memoria RAM: Los requisitos de memoria RAM dependen en gran medida del tamaño de tu base de datos y de la cantidad de tráfico que esperas en tu sitio web. Para un entorno de desarrollo o pruebas, 1 GB de RAM puede ser suficiente. Sin embargo, en un entorno de producción con una gran cantidad de datos y tráfico, es posible que necesites varios gigabytes de RAM.
4. Procesador: phpMyAdmin no es especialmente exigente en términos de poder de procesamiento. Un procesador de un solo núcleo y de velocidad moderada suele ser suficiente para ejecutarlo sin problemas. Sin embargo, si esperas un alto volumen de tráfico o estás realizando operaciones intensivas en la base de datos, un procesador más potente puede ser beneficioso.

Software

* Servidor web: phpMyAdmin es una aplicación basada en web, por lo que necesitarás tener un servidor web en funcionamiento, como Apache o Nginx. Asegúrate de que el servidor web esté correctamente configurado y funcionando antes de instalar phpMyAdmin.
* PHP: phpMyAdmin está escrito en PHP, por lo que necesitarás tener PHP instalado en tu servidor. phpMyAdmin es compatible con PHP 7.1 o superior, pero se recomienda usar la versión más reciente disponible para obtener las últimas mejoras de seguridad y rendimiento. Asegúrate de tener las extensiones PHP necesarias habilitadas, como la extensión de MySQLi.
* Servidor de base de datos: phpMyAdmin se utiliza para administrar bases de datos MySQL o MariaDB. Por lo tanto, necesitarás tener un servidor de base de datos instalado y configurado. Puedes usar MySQL o MariaDB según tus preferencias y requisitos. Asegúrate de tener las credenciales de acceso al servidor de base de datos disponibles para configurar phpMyAdmin correctamente.
* Sistema operativo: phpMyAdmin es compatible con una amplia gama de sistemas operativos, incluidos Windows, macOS y varias distribuciones de Linux. Asegúrate de que tu sistema operativo cumpla con los requisitos mínimos para phpMyAdmin.
* Navegador web: Para acceder a la interfaz de phpMyAdmin, necesitarás un navegador web compatible. phpMyAdmin es compatible con la mayoría de los navegadores web modernos, como Chrome, Firefox, Safari, etc.

# Listado de requisitos para el lenguaje de interfaz

Hardware

1. Servidor web: Necesitarás tener un servidor web en funcionamiento, como Apache o Nginx. Asegúrate de que el servidor web esté correctamente configurado y funcionando antes de instalar PHP.
2. Sistema operativo: PHP es compatible con una amplia gama de sistemas operativos, incluyendo Windows, macOS y varias distribuciones de Linux. Asegúrate de que tu sistema operativo cumpla con los requisitos mínimos para PHP.
3. Espacio en disco: PHP en sí mismo no requiere mucho espacio en disco. Sin embargo, necesitarás espacio suficiente para alojar tu servidor web, las aplicaciones y cualquier otro componente necesario para tu sitio web o aplicación.
4. Memoria RAM: Los requisitos de memoria RAM para PHP pueden variar dependiendo del tamaño y la complejidad de tu aplicación. Para un entorno de desarrollo o pruebas, 1 GB de RAM puede ser suficiente. Sin embargo, en un entorno de producción con una gran cantidad de tráfico o aplicaciones complejas, es posible que necesites varios gigabytes de RAM.
5. Procesador: PHP no es particularmente exigente en términos de poder de procesamiento. Un procesador de un solo núcleo y velocidad moderada suele ser suficiente para ejecutarlo sin problemas. Sin embargo, si esperas un alto volumen de tráfico o estás ejecutando aplicaciones intensivas en términos de procesamiento, un procesador más potente puede ser beneficioso.

Software

* Servidor web: Necesitarás tener un servidor web instalado y configurado en tu servidor. Los servidores web populares son Apache, Nginx e IIS. Asegúrate de que el servidor web esté funcionando correctamente antes de instalar PHP.
* Sistema operativo: PHP es compatible con una amplia gama de sistemas operativos, incluyendo Windows, macOS y varias distribuciones de Linux. Asegúrate de que tu sistema operativo sea compatible con la versión de PHP que deseas instalar. Puedes consultar la documentación oficial de PHP para verificar la compatibilidad.
* Base de datos: Si tu aplicación web requiere acceso a una base de datos, necesitarás instalar y configurar un sistema de gestión de bases de datos compatible, como MySQL, MariaDB o PostgreSQL. Asegúrate de que la base de datos esté funcionando correctamente antes de instalar PHP.
* PHP: Por supuesto, necesitarás descargar e instalar la versión de PHP que deseas utilizar. Puedes obtener la última versión de PHP en el sitio web oficial de PHP (www.php.net). Asegúrate de descargar la versión adecuada para tu sistema operativo y seguir las instrucciones de instalación proporcionadas en la documentación de PHP.
* Dependencias y extensiones: PHP puede requerir ciertas dependencias y extensiones para funcionar correctamente. Algunas extensiones comunes son MySQLi (para trabajar con MySQL/MariaDB), PDO (PHP Data Objects), GD (para manipulación de imágenes) y OpenSSL (para cifrado). Asegúrate de instalar y configurar las extensiones necesarias según los requisitos de tu aplicación.

Manual técnico Diccionario de datos Tabla: **artículos**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_producto | int(11) | clave primaria. Identificador único del producto. |
| id\_categoria | int(11) | Identificador de la categoría a la que pertenece el producto. |
| id\_imagen | int(11) | Identificador de la imagen asociada al producto. |
| id\_usuario | int(11) | Identificador del usuario que capturó el producto. |
| nombre | varchar(50) | Nombre del producto. |
| descripcion | varchar(500) | Descripción del producto. |
| cantidad | int(11) | Cantidad disponible del producto. |
| precio | float | Precio del producto. |
| fechaCaptura | date | Fecha en la que se capturó el producto. |

Tabla: **categorías**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_categoria | int(11) | clave primaria. Identificador único de la categoría. |
| id\_usuario | int(11) | Identificador del usuario al que pertenece la categoría. |
| nombreCategoria | varchar(150) | Nombre de la categoría. |
| fechaCaptura | date | Fecha en la que se capturó la categoría. |

Tabla: **clientes**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_cliente | int(11) | clave primaria. Identificador único del cliente. |
| id\_usuario | int(11) | Identificador del usuario al que pertenece el cliente. |
| nombre | varchar(200) | Nombre del cliente. |
| apellido | varchar(200) | Apellido del cliente. |
| direccion | varchar(200) | Dirección del cliente. |
| email | varchar(200) | Email del cliente. |
| telefono | varchar(200) | Teléfono del cliente. |
| rfc | varchar(200) | RFC del cliente. |

Tabla: **imágenes**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_imagen | int(11) | Clave primaria. Identificador único de la imagen. |

|  |  |  |
| --- | --- | --- |
| id\_categoria | int(11) | Identificador de la categoría a la que pertenece la imagen. |
| nombre | varchar(500) | Nombre de la imagen |
| ruta | varchar(500) | Ruta de la imagen en el disco. |
| fechaSubida | date | Fecha en la que se subió la imagen. |

Tabla: **usuarios**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_usuario | int(11) | clave primaria. Identificador único del usuario. |
| nombre | varchar(50) | Nombre del usuario. |
| apellido | varchar(50) | Apellido del usuario. |
| email | varchar(50) | Email del usuario. |
| password | tinytext | Contraseña del usuario. |
| fechaCaptura | date | Fecha en la que se capturó el usuario. |

Tabla: **ventas**

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| id\_venta | int(11) | Clave primaria. Identificador único de la venta. |
| id\_cliente | int(11) | Identificador del cliente al que le pertenece la venta |
| id\_producto | int(11) | Identificador del producto al que le pertenece la venta |
| id\_usuario | int(11) | Identificador del usuario al que le pertenece la venta |
| precio | float | Precio total de la venta |
| fechaCompra | date | Fecha en la que se realizó la venta |

Fsf

S

Fsd

Fds

fsds

# Listado de explicación:

## Vistas

1. Vista vista\_ventas: CREATE VIEW vista\_ventas AS

SELECT v.id\_venta, c.nombre AS nombre\_cliente, p.nombre AS nombre\_producto, u.nombre AS nombre\_usuario, v.precio, v.fechaCompra

FROM ventas v

INNER JOIN clientes c ON v.id\_cliente = c.id\_cliente INNER JOIN productos p ON v.id\_producto = p.id\_producto INNER JOIN usuarios u ON v.id\_usuario = u.id\_usuario;

Esta vista combina información de las tablas ventas, clientes, productos y usuarios para proporcionar una vista más legible de los datos de venta. La vista incluye los siguientes campos: id\_venta, nombre\_cliente, nombre\_producto, nombre\_usuario, precio y fechaCompra.

1. Vista vista\_categorias: CREATE VIEW vista\_categorias AS

SELECT c.id\_categoria, c.nombreCategoria, COUNT(a.id\_producto) AS total\_productos FROM categorias c

LEFT JOIN articulos a ON c.id\_categoria = a.id\_categoria GROUP BY c.id\_categoria, c.nombreCategoria;

Esta vista muestra un resumen de las categorías de productos junto con la cantidad total de productos en cada categoría. La vista incluye los campos: id\_categoria, nombreCategoria y total\_productos.

## Usuarios y Permisos

1. En este punto los usuarios que serian nuestros empleados solo tienen el permiso de agregar artículos y categorías, así como ver las ventas hechas y generarlas.
2. Así como agregar también clientes y sus datos

## Store Procedure

1. articulos\_DATOSGRAFICO\_BAR

DROP PROCEDURE IF EXISTS `articulos\_DATOSGRAFICO\_BAR`; CREATE PROCEDURE `articulos\_DATOSGRAFICO\_BAR` () BEGIN

SELECT \* FROM articulos; END

Este procedimiento devuelve todos los registros de la tabla artículos.

## Funciones

1. Función definida calcular\_total\_venta:

CREATE FUNCTION calcular\_total\_venta(venta\_id INT) RETURNS DECIMAL(10,2) BEGIN

DECLARE total DECIMAL(10,2);

SELECT SUM(precio) INTO total FROM ventas

WHERE id\_venta = venta\_id;

RETURN total; END

Esta función toma como parámetro el id\_venta y calcula el total de la venta sumando los precios de todos los productos asociados a esa venta en la tabla ventas.

## Triggers

1. Disparador actualizar\_stock\_ventas:

CREATE TRIGGER actualizar\_stock\_ventas AFTER INSERT ON ventas FOR EACH ROW

BEGIN

UPDATE articulos

SET cantidad = cantidad - 1

WHERE id\_producto = NEW.id\_producto; END

Este disparador se ejecuta después de insertar un nuevo registro en la tabla ventas. Actualiza el campo cantidad en la tabla artículos, reduciendo en 1 la cantidad disponible del producto vendido.

1. Disparador actualizar\_precio\_categoria:

CREATE TRIGGER actualizar\_precio\_categoria AFTER UPDATE ON articulos FOR EACH ROW

BEGIN

UPDATE categorias c

INNER JOIN articulos a ON c.id\_categoria = a.id\_categoria

SET c.precio\_promedio = (SELECT AVG(precio) FROM articulos WHERE id\_categoria = c.id\_categoria)

WHERE a.id\_categoria = NEW.id\_categoria; END

Este disparador se ejecuta después de actualizar un registro en la tabla articulos. Actualiza el campo precio\_promedio en la tabla categorias, calculando el promedio de precios de los productos pertenecientes a la misma categoría.